

REMARKS

The Examiner is thanked for the performance of a thorough search.

Prior to entry of this response, Claims 1-41 were pending in the application. By this response, no claims are added or canceled. Hence, Claims 1-41 are pending in the application upon entry of this response.

Claims 1, 2, 4-6, 8-13, and 15-41 are amended herein.

SUMMARY OF REJECTIONS

Claims 21-40 were rejected under 35 U.S.C. 101 as allegedly directed to non-statutory subject matter.

Claims 2-10, 14-15, 17-20, 22-30, 34-35 and 37-40 were rejected under 35 U.S.C. 112, second paragraph, for alleged indefiniteness.

Claims 1-41 were rejected under 35 U.S.C. 103(a) as allegedly unpatentable over Reid et al. (U.S. Patent No. 6,560,592, hereafter referred to as "*Reid*"), in view of Detlef Zimmer et al. ("On the Semantics of Complex Events in Active Databases Management Systems", Proc. Of the 15th Int'l Conf. on Data Engineering, hereafter referred to as "*Zimmer*").

THE REJECTIONS NOT BASED ON PRIOR ART

Rejections under 35 USC §101 - Claims 21-40

Claims 21-40 are rejected under 35 U.S.C. 101 as allegedly directed to non-statutory subject matter. Specifically, these computer-readable medium claims were rejected because the specification describes computer-readable media as including a carrier wave and transmission media.

Claims 21-40 are amended herein to replace “computer-readable medium” with “computer-readable storage medium”, thereby clearly placing the subject matter of these claims under the statutory category of “a manufacture.” Reconsideration and withdrawal of the rejection of Claims 21-40 under 35 U.S.C. §101 is requested.

Rejections under 35 USC §112 - Claims 2-10, 14-15, 17-20, 22-30, 34-35, 37-40

Claims 2-10, 14-15, 17-20, 22-30, 34-35 and 37-40 are rejected under 35 U.S.C. 112, second paragraph, as allegedly indefinite.

Regarding **Claim 9**, the Office Action asserts that there is insufficient antecedent basis for use of the clause “storing said expressions in columns of a table.” This rejection is understood to be based on insufficient antecedent basis for the phrase “expressions in columns of a table” because the limitation in Claim 1 to which this clause refers was “storing said expression *in a table...*” rather than “in *columns of a table.*”

Claim 1 is amended herein to recite:

storing said event structure, said first set of one or more conditions, and said action preferences in said database;

and Claim 9 is amended herein to recite:

wherein storing comprises storing one or more conditions from said first set as an EXPRESSION data type in an EXPRESSION column of a database table.

These amendments overcome this rejection as understood. Reconsideration and withdrawal of the rejection of Claim 9 under 35 U.S.C. §112, second paragraph, is requested.

Regarding **Claims 2-8, 10, 14-15, 17-20, 22-30, 34-35 and 37-40**, the Office Action asserts that each of these claims is improper because each attempts to over-write or replace a limitation in a parent claim. This rejection is traversed, based on the following reasoning.

Respectfully, these claims do not attempt to over-write or replace a limitation in the parent claim. Rather, these claims augment and further limit corresponding limitations in the parent claim. There are really only two kinds of valid dependent claims: those that add new limitations and those that further limit existing limitations. Claim 2 happens to be a valid dependent claim because it further limits an existing limitation from Claim 1.

Regarding Claim 3, for example, the wording “receiving an expression that identifies said event structure as a composite event structure having two or more primitive events that are each represented, in said database, as an object type embedded in said composite event structure” does not over-write or replace the limitation in Claim 1 reciting “receiving an expression that identifies an event structure, ... , wherein said event structure defines an event that corresponds with said event structure by defining a set of attributes that describe features of a corresponding event” Rather, Claim 3 further limits what is received in Claim 1, by further limiting the event structure that is received in Claim 1 to an event structure defined as a composite event structure having two or more primitive events that are each represented, in said database, as an object type embedded in said composite event structure. Thus, the act of receiving a specific form of event structure as recited in Claim 3 is subsumed by the act of receiving an event structure, generally, as recited in Claim 1. Hence, Claim 3 does not contradict or relate inconsistently with Claim 1, and does not include an independent limitation that supplants a limitation in Claim 1. The step of receiving recited in Claim 3 further limits a corresponding step of receiving recited in Claim 1 without contradicting or otherwise being inconsistent with the corresponding step of Claim 1. Consequently, a party infringing Claim 3 would also infringe Claim 1, but not vice versa.

The foregoing discussion in reference to dependent Claim 3 applies also to the other dependent claims rejected under 35 U.S.C. 112, second paragraph. That is, the rejected

limitations recited in these dependent claims further limit, and do not contradict, corresponding limitations recited in the parent Claim 1.

Based on the foregoing, it is shown that each of Claims 2-8, 10, 14-15, 17-20, 22-30, 34-35 and 37-40 is presented in a proper claim format and is readily understandable. Therefore, each of these claims is not indefinite for failing to particularly point out and distinctly claim the subject matter. Reconsideration and withdrawal of the rejection of Claims 2-8, 10, 14-15, 17-20, 22-30, 34-35 and 37-40 under 35 U.S.C. §112, second paragraph, is respectfully requested.

THE REJECTIONS BASED ON PRIOR ART

Rejections under 35 USC 103(a) - Claims 1-41

Claims 1-41 were rejected under 35 U.S.C. 103(a) as allegedly unpatentable over *Reid* in view of *Zimmer*. However, there are several fundamental differences between these claims and the teachings of *Reid* and *Zimmer*, as described hereafter.

Claim 1 recites:

receiving an expression that identifies an event structure, a first set of one or more conditions related to said event structure, and one or more action preferences related to said event structure, wherein said event structure defines an event that corresponds with said event structure by defining a set of attributes that describe features of a corresponding event;
 storing said event structure, said first set of one or more conditions, and said action preferences in said database;
 during a database session,
 receiving a first event,
 detecting that said first event is an occurrence of said event by comparing said first event to said event structure and determining that said first event corresponds with said event structure,

based on said detecting, selecting said first set of one or more conditions for evaluation against said first event, and determining whether said first event satisfies any of said one or more conditions in said first set...

Thus, Claim 1 involves receiving and storing an expression (e.g., a SQL expression) that identifies **an event structure that defines a corresponding event by defining a set of attributes that describe features of a corresponding event**. The event structure describes, on an abstract level, the essential factors that unambiguously identify the occurrence of an event of that type (see, e.g., paragraph [0023]). Hence, the method recited in Claim 1 includes **detecting that a particular event is in fact an occurrence of an event defined by the event structure, by comparing the particular event with the event structure**. Before identifying, selecting, and evaluating conditions relating to a particular type of event, it first needs to be determined **of what event type a particular event is**. Resources would be wasted by trying to evaluate conditions in view of inapplicable events, i.e., events of a different type than the type of event to which the conditions apply. Such determination is made based on comparing the event structure with the particular event. Thus, once it is determined of what event type a particular event is, then **the corresponding conditions can be identified and selected from storage in the database, for evaluation against the event**. None of the cited references teach or fairly suggest this type of functionality in a database-centric rules engine.

The Office Action asserts that *Zimmer* teaches receiving the expression that identifies an event structure, by vaguely discussing that “events that cause a complex event to occur ... provide *all information* about the event that is necessary for an appropriate execution of the other parts of the rule” (emphasis added; page 392, fourth paragraph of ‘Introduction’). This

citation from *Zimmer* does not teach or suggest the notion of an event structure that defines an event that corresponds with the event structure by defining a set of attributes that describe features of a corresponding event, i.e., defines an event type. The statement that such events provide all information about the event necessary for execution of other parts of the rule is interpreted as meaning that such events comprise information that is to be evaluated in view of a condition corresponding to the rule. This is not the same as an event structure that *defines* a corresponding event so that such an event can be recognized by the database system, but rather refers to the event payload that is used to evaluate a corresponding condition.

The Office Action asserts that *Reid* teaches storing the expression in the database. However, *Reid* may store rulesets in a database, but the rulesets of *Reid* do not include the event structure as recited in Claim 1. With no event structures being stored in *Reid*, or in *Zimmer*, it is not possible to perform the real-time detection of particular events by comparing incoming events with one or more event structures stored in the database.

The Office Action asserts that *Reid* teaches detecting an occurrence of an event by detecting when an event occurs that complies with the corresponding event structure, by discussing the trigger that causes evaluation of a premise. However, Reid states that “the trigger causes evaluation of the premise upon the occurrence of a predetermined event” (Abstract). An event triggering evaluation of a condition is known in the context of rules engines, and it is conceded that *Reid* and *Zimmer* discuss such. However, neither *Reid* nor *Zimmer* teaches or suggests the event structure mechanism recited in Claim 1 for determining whether any given event is in fact an event that is to be evaluated against a particular set of condition(s).

By contrast, *Reid* discusses occurrence of a **predetermined event** that triggers evaluation of a premise. Also by contrast, *Zimmer* discusses that primitive events that cause a complex event to occur are *bound* to the complex event, which presumably means that events are also bound to corresponding rule conditions. Hence, neither *Reid* nor *Zimmer* teaches or fairly suggests a database rules engine that is functionally capable of storing many different rulesets that are not pre-bound to predetermined events, but which contain conditions that are selected for evaluation at runtime in response to determining the types of arbitrary events that occurred based on stored event structures and, therefore, determining the particular conditions that are to be evaluated against various events. The triggering events of *Reid* and *Zimmer* may cause evaluation of a premise/condition, but such triggering events do not detect that a particular event corresponds to a particular event structure, and therefore is to be evaluated in the context of a corresponding set of condition(s), by comparing the particular event with a stored event structure that defines the event type.

The foregoing discussion shows that no combination of the cited references of *Reid* and *Zimmer* teaches or suggests each and every limitation of Claim 1. Hence, no possible combination of *Reid* and *Zimmer* would result in the embodiment recited in Claim 1. Therefore, one of ordinary skill in the art would not be able to derive the embodiment of Claim 1 from the cited references and, consequently, this embodiment is not obvious in view of the cited references.

Independent **Claim 41** is a system claim, in which are recited elements based on similar subject matter as in Claim 1. Therefore, Claim 41 is patentable over the cited references of record for at least the same reasons discussed herein in reference to Claim 1. Dependent **Claims 2-40** depend from Claim 1 and, therefore, are patentable over the cited references of record for at least the same reasons discussed herein in reference to Claim 1.

Furthermore, each of these dependent claims includes at least one other limitation that makes it further patentable over the references of record. However, due to the fundamental difference between Claim 1 and *Reid* and *Zimmer* discussed above, discussion of these additional differences is unnecessary and is foregone at this time beyond the extent that may be presented hereafter. However, the rejection of the dependent claims is collectively traversed, and no statements of official notice or overarching allegations of obviousness that may be present in the Office Action are stipulated to, no allegations of well-known features that may be present in the Office Action are admitted as prior art features, and the right to separately argue such statements and features in the future is not disclaimed. Reconsideration and withdrawal of the rejections of Claims 1-41 under 35 U.S.C. §103(a) is respectfully requested.

For example, **Claim 4** is in the context of composite (e.g., complex) events consisting of multiple primitive events. It is acknowledged that *Zimmer* is directed to a similar context of processing complex events in a database management system. However, Claim 1 recites **persistently storing results about whether an occurrence of a first primitive event satisfies any conditions** in an expression or rule corresponding to the composite event. The Office Action relies on the ruleset and database of *Reid* for an alleged teaching of this element. However, storing a ruleset in a database does not mean that intermediate results from conditions on primitive events of a composite event are stored in the database. Further, neither *Reid* nor *Zimmer* explicitly teaches, suggests, or implies that such intermediate results are stored persistently in the database rather than, for example, in local volatile memory. For at least this reason, Claim 4 is further patentable over the references of record.

For example, **Claim 10** recites that the received expression, which essentially characterizes a rule, is represented as a SQL query on a database (see, e.g., paragraph

[0082]). The Office Action relies on the triggers that are setup to be executed in response to various database operations, in *Reid*, for an alleged teaching of this element. However, triggers which are executed in response to certain database operations, or perhaps even in response to SQL commands corresponding to such operations, is not the same as receiving a SQL expression that embodies an entire ECA (Event-Condition-Action) rule which includes an event type definition, etc. As described in paragraph [0016] of the application, embodiments of the rules engine described therein departs from existing rules engines by managing ECA rules in a relational database, where the events for which the rules are defined are relational in nature and a subset of the standard structured query language (SQL) can be used to identify the occurrence of interesting events, and to evaluate conditions against the event occurrences. Neither *Reid* nor *Zimmer* explicitly teaches, suggests, or implies that **rules are defined and stored in a database via SQL expressions**. For at least this reason, Claim 10 is further patentable over the references of record.

For example, **Claim 12** recites the use of a database view for tracking various events that have satisfied various conditions, and corresponding action preferences. A database view is, effectively, a reusable query used to extract specific information from a database and present such information to a user. As discussed in paragraphs [0072] and [0073] of the application, the capability to present the rule session results in a view allows users to perform additional operations on the results, as a set, and thus identify a subset of the conditions for action execution. Hence, the rule set results view can be used to support complex conflict resolution criteria among matching conditions. The results from the rule set results view can further be used to schedule actions outside the database. The Office Action relies on a neural network of *Reid* for an alleged teaching of this element. However, **a neural network of rules is not the same as a database view**, nor do they perform similar functions.

Furthermore, neither *Reid* nor *Zimmer* actually teaches or suggests using a database view for the specific purpose recited in Claim 12. For at least this reason, Claim 12 is further patentable over the references of record.

For example, **Claim 17** recites receiving an expression that identifies a temporal condition, where a temporal condition specifies that **an associated action is to be performed if a second condition is satisfied by an occurrence of an event, within a particular time after a first condition from set of conditions is satisfied by an occurrence of an event**. It is acknowledged that *Zimmer* mentions “temporal event types.” However, Claim 17 recites a specific meaning for a temporal condition, which is not reflected in the description of temporal event types in *Zimmer*. As discussed in paragraph [0041] of the application, rules involving temporal events are activated when an event is detected or not detected within a specified timeframe. An example of a rule involving temporal events is as follows: if an order is placed by a Gold customer and the ordered items are shipped within 12 hours of the order placement, then increment quality of service statistics. It is further noted that a temporal-type condition is not the same as a sequencing-type condition, as discussed in *Zimmer* and in paragraphs [0053] and [0054] of the application. Thus, neither *Reid* nor *Zimmer* actually teaches or suggests using temporal conditions, as explicitly recited in Claim 17. For at least this reason, Claim 17 is further patentable over the references of record.

For example, **Claim 19** recites receiving an expression (an “any *n*” condition) that identifies a group of conditions which, when a particular number of conditions from the group of conditions is satisfied by one or more occurrences of events, triggers performance of a corresponding action, wherein the particular number is less than a number of conditions in said group of conditions. As discussed in paragraph [0049] and [0050] of the application,

the actions that are associated with conditions using such constructs are performed when any n events of the specified events are detected. An example of a rule involving "any n " is as follows: if a customer adds two of the following items to a shopping cart, then suggest a tripod to the customer: a camcorder lens worth more than \$100, a lens filter, and an IR light. The Office Action relies on the group of event instances in the "Example 3" of *Zimmer*. However, the referenced example does not include an "any n " type of condition, but is more akin to a sequencing condition. It is further noted that a sequencing-type condition is not the same as an "any n " type of condition. Thus, neither *Reid* nor *Zimmer* actually teaches or suggests using "any n " conditions, as explicitly recited in Claim 19. For at least this reason, Claim 19 is further patentable over the references of record.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims are now in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER
LLP

Date: 1/26/07

John D. Henkhaus
John D. Henkhaus
Reg. No. 42,656

2055 Gateway Place, Suite 550
San Jose, CA 95110-1089
(408) 414-1080
Facsimile: (408) 414-1076

OID 2003-248-01

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

on 1/26/07 by Martina Placid
22 Martina Placid